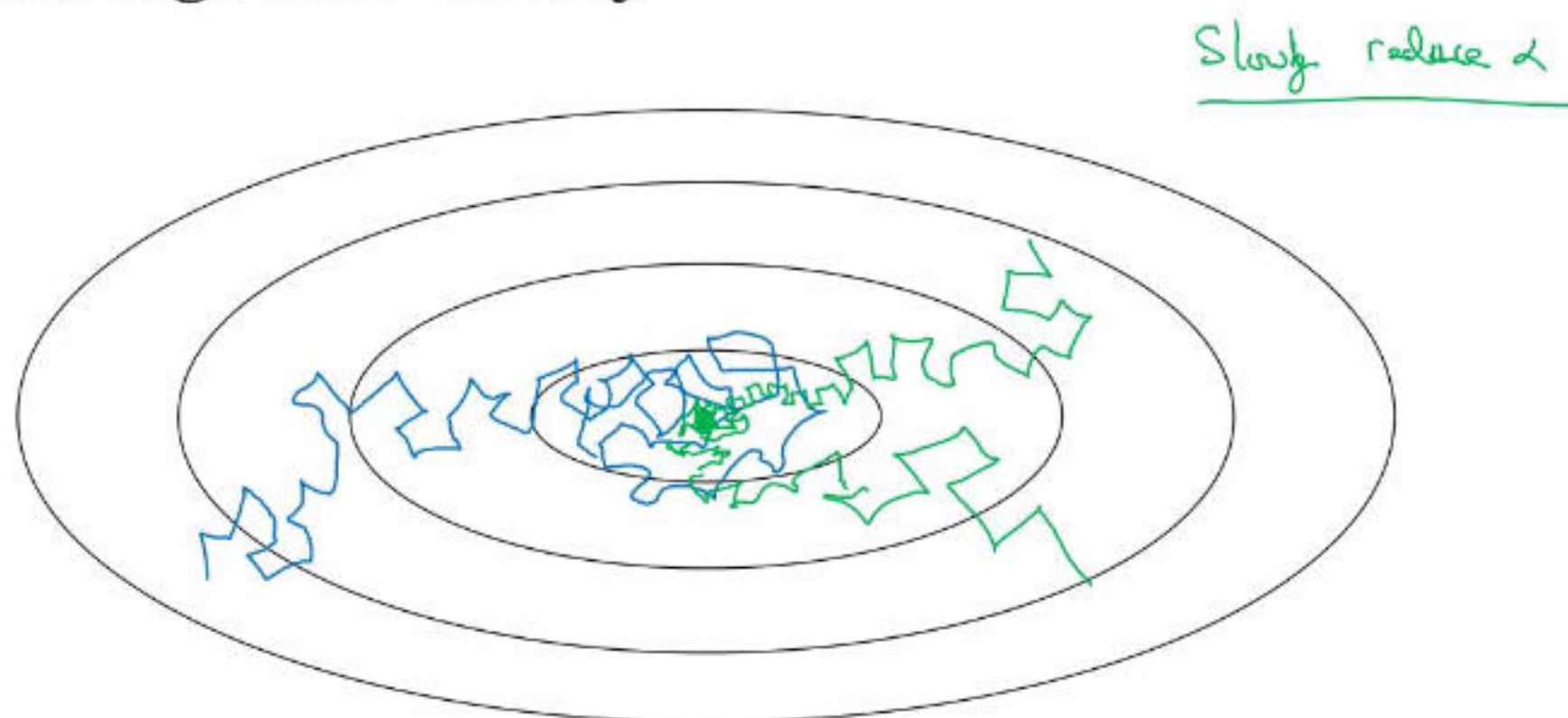


12 20

Learning rate Decay.

Learning rate decay



Andrew Ng

α 随 epoch \uparrow 逐渐减小, 在最小值的小邻域内波动。
常见公式: $\alpha = \frac{1}{1 + \text{decay-rate} * \text{epoch}} \alpha_0$

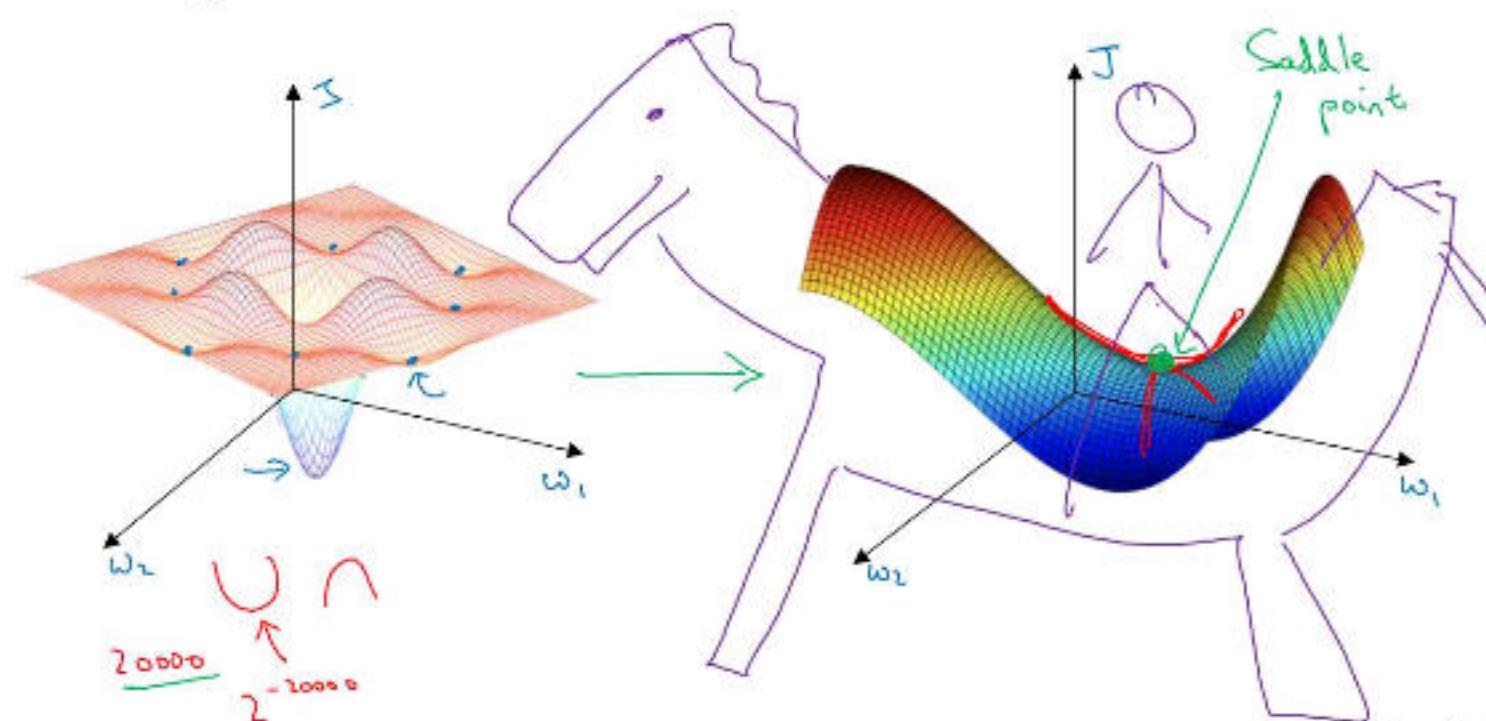
$$\alpha = 0.95^{\text{epoch}} * \alpha_0$$

$$\alpha = \frac{k}{\sqrt{\text{epoch}}} * \alpha_0 \text{ or } \frac{k}{\sqrt{t}} * \alpha_0$$

Local optima.

高维空间不容易出现局部最优,
多为鞍点。

Local optima in neural networks



Andrew Ng

鞍点减缓学习过程, 可以通过
mini-batch 扰动摆脱,
很慢
。。

12 20.

Hyper Param Tune.

Prior:

1. α
 2. β , # hidden-units, batch size.
 3. # layers, learning rate decay,
 4. $\beta_1, \beta_2, \epsilon$.
- 0.9 0.999 10^{-8} .

加权平均.

调参方法: 1. 随机采样 2. 由粗到精.

3. baby sit one model ~ Panda.
Train models in parallel ~ Caviar.

Batch 归一化:

对于 x_i , 做归一化.

类似地, 对 $z^{(1)} \dots z^{(l-1)}$ 也做归一化.

$$\mu = \frac{1}{M} \sum_i z^{(l)}$$

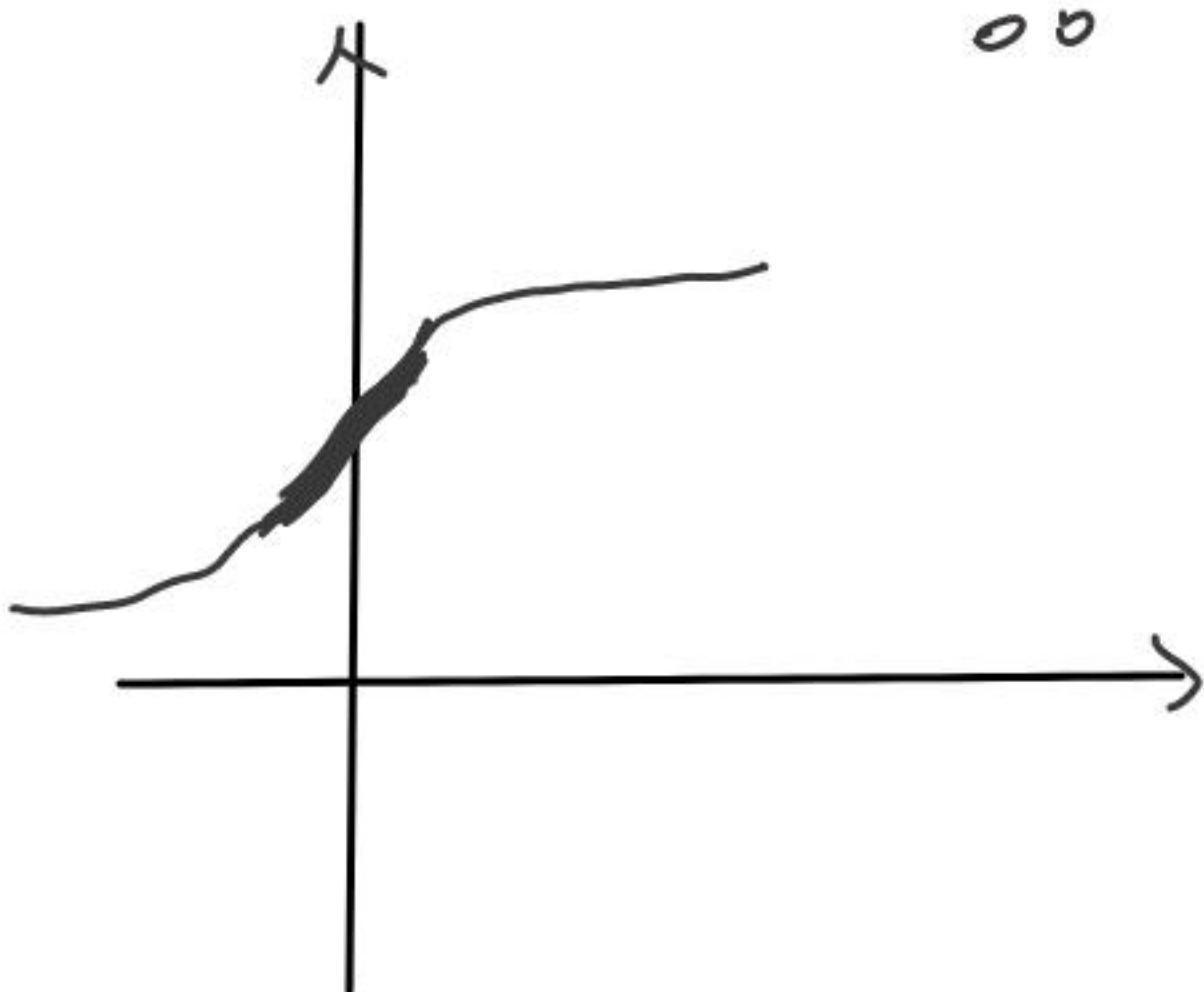
$$z^{(l)}_{\text{norm}} = \frac{z^{(l)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

$$\sigma^2 = \frac{1}{M} \sum_i (z^{(l)} - \mu)^2$$

$$\tilde{z}^{(l)} = \gamma z^{(l)}_{\text{norm}} + \beta.$$

β, γ 用于调整方差与均值.

不希望 $z^{(l)}$ 总是输入线性激活函数.



12 21

应用到 NN:

$$z^{[1]} \xrightarrow{\beta^{[1]}, \gamma^{[1]}} \tilde{z}^{[1]} \longrightarrow a^{[1]} \longrightarrow \dots$$

$\beta^{[1]}, \gamma^{[1]}$ 也视作参数进行 Gradient Descent.

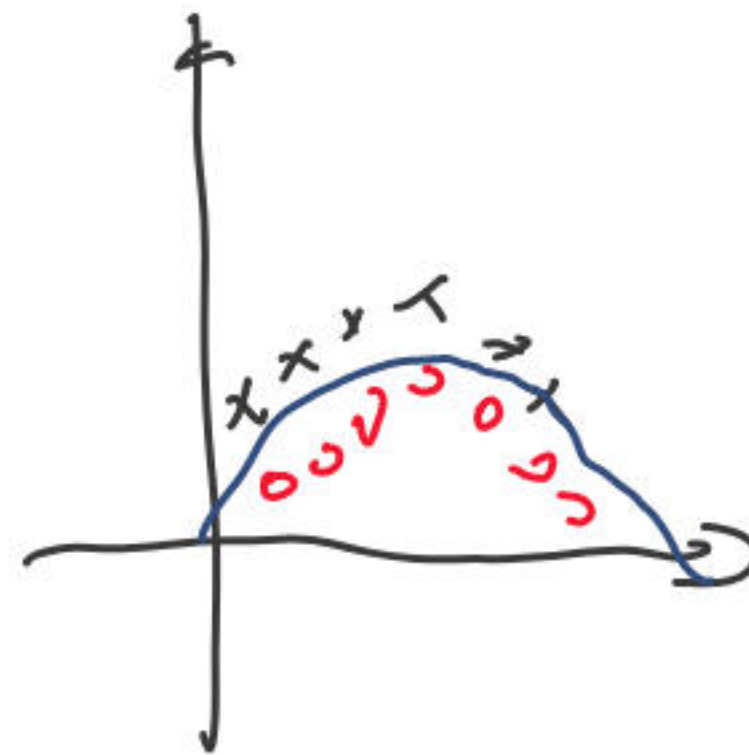
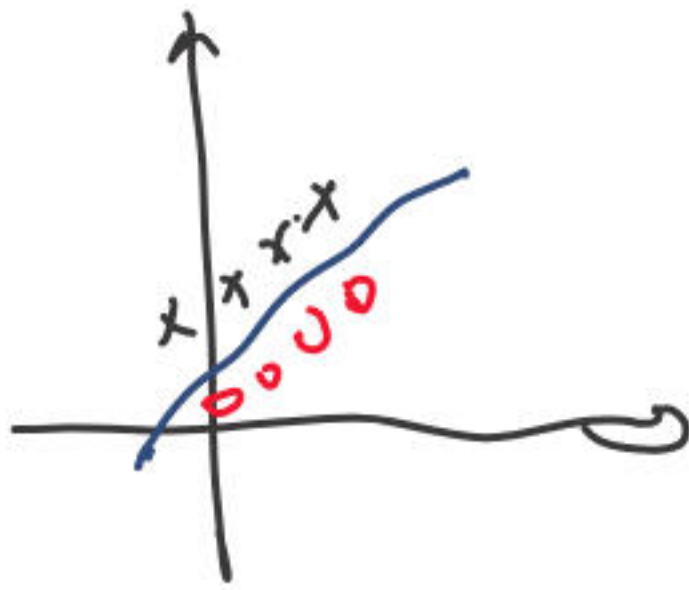
$$z^{[1]} = w^{[1]} a^{[0]} + \cancel{b^{[1]}}$$

$$A^{[1]} = g(\tilde{z}^{[1]}).$$

$$\tilde{z}^{[1]} = \gamma^{[1]} z_{\text{norm}}^{[1]} + \beta^{[1]}$$

Params: $w^{[1]}, \beta^{[1]}, \gamma^{[1]}$.
($n_1 \times 1$)

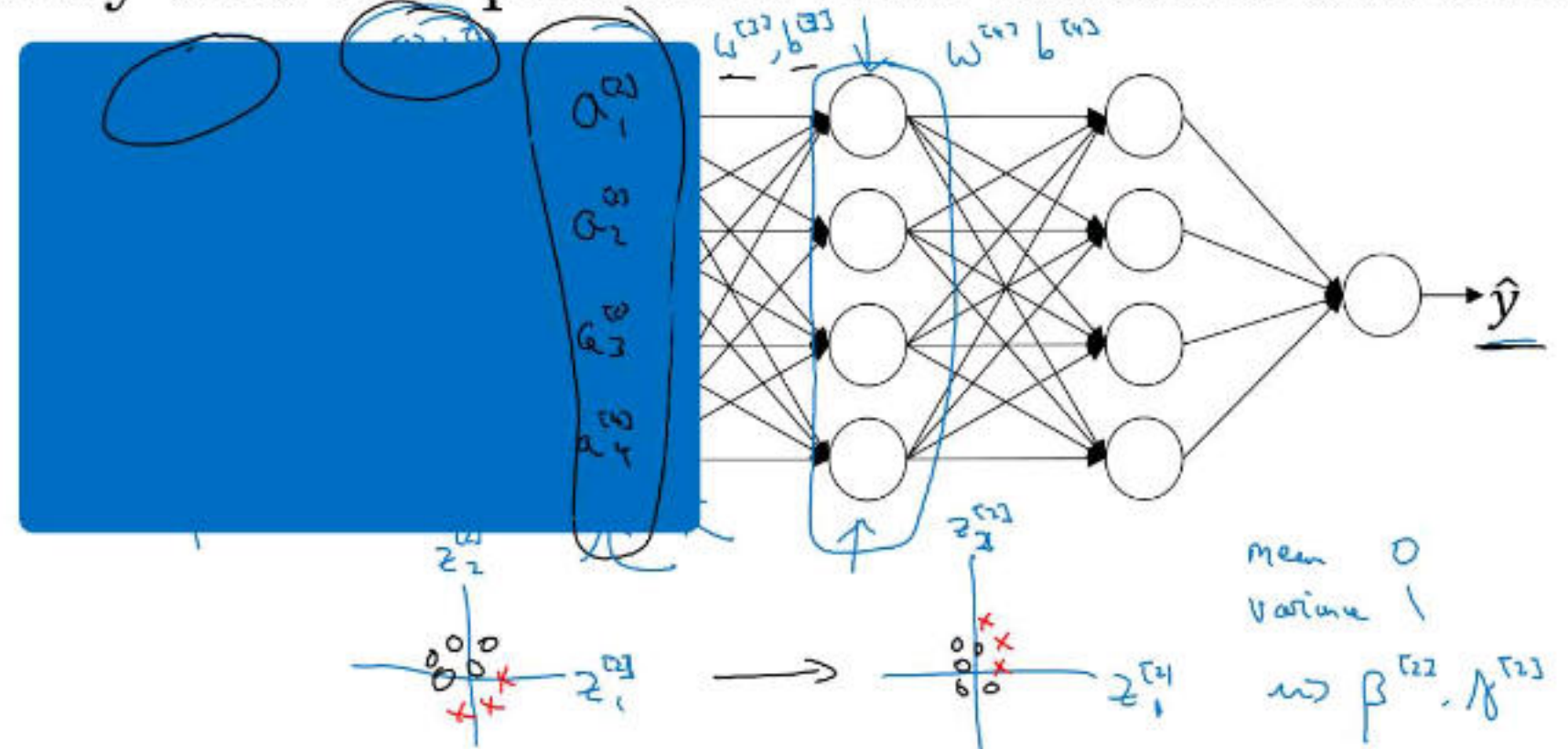
Intuition:



Data 改变了, 应重新训练 Model,

$a^{[1]}$ 变动, Model
重新训练, 归一化便
变动变小, 加速训练.

Why this is a problem with neural networks?



12 21.

μ, σ^2 是在 batch 上计算的有噪声, 有一定 Reg 效果, 测试时样本不多, 不能算 μ, σ^2 .

在训练集上对 $\mu^{(l)}$ 做指数加权平均.

Soft Max.

$C = \# \text{ classes.}$

$n_l = 4.$

$a^{[l]} = (4, 1).$

$$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}.$$

激活函数: $t = e^{z^{[l]}}$

$$z^{[l]} = (4, 1), \quad e^{z^{[l]}} = (4, 1).$$

$$a_i^{[l]} = \frac{t_i}{\sum_j t_j}$$

Hard Max

VS

Soft Max.

$$\begin{matrix} \text{Hard Max} \\ \downarrow \\ [1 \quad 0 \quad 0 \quad 0] \end{matrix}$$

$$\begin{matrix} \text{Soft Max} \\ \downarrow \\ [0.8 \quad 0.1 \quad 0.1] \end{matrix}$$

若 $C=2$, SoftMax \Leftrightarrow Logistic.

$$L(\hat{y}, y) = - \sum_{j=1}^4 y_j \log \hat{y}_j$$

$$J(w^{[l]}, b^{[l]}) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

Back Prop: $\frac{\partial J}{\partial z^{[l]}} = \hat{y} - y.$

12 21

Tensor Flow.

通过前向传播绘制计算 Graph. 自动反向传播.

